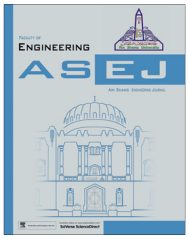
Ain Shams Engineering Journal (2013) 4, 753–762

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com



ELECTRICAL ENGINEERING

A hybrid network intrusion detection framework based on random forests and weighted k-means

Reda M. Elbasiony *, **Elsayed A. Sallam** [1], **Tarek E. Eltobely** [2], **Mahmoud M. Fahmy** [3]

Tanta University, Faculty of Engineering, Tanta, Gharbia, Egypt

Received 7 August 2012; revised 2 January 2013; accepted 20 January 2013
Available online 7 March 2013

KEYWORDS

Computer network security;
Data mining;
Intrusion detection;
Random forests;
k-Means

Abstract Many current NIDSs are rule-based systems, which are very difficult in encoding rules, and cannot detect novel intrusions. Therefore, a hybrid detection framework that depends on data mining classification and clustering techniques is proposed. In misuse detection, random forests classification algorithm is used to build intrusion patterns automatically from a training dataset, and then matches network connections to these intrusion patterns to detect network intrusions. In anomaly detection, the k-means clustering algorithm is used to detect novel intrusions by clustering the network connections' data to collect the most of intrusions together in one or more clusters. In the proposed hybrid framework, the anomaly part is improved by replacing the k-means algorithm with another one called weighted k-means algorithm, moreover, it uses a proposed method in choosing the anomalous clusters by injecting known attacks into uncertain connections data. Our approaches are evaluated over the Knowledge Discovery and Data Mining (KDD'99) datasets.

1. Introduction

Computer networks have become widely ubiquitous, used to transfer a lot of sensitive information between many types of computer devices, from huge servers to mobile devices and minicomputers. Although many types of security methods, like access control, encryption, and firewalls, are used, network security breaches increase day by day [1]. So, there is an urgent need to intelligent intrusion detection systems (IDSs) to detect novel intrusions automatically. There are two major intrusion detection methods: misuse detection and anomaly detection. Misuse detection compares network activities with pre-defined signatures or patterns taken from characteristic features that

* Corresponding author. Tel.: +20 1002937033.
E-mail addresses: reda@f-eng.tanta.edu.eg (R.M. Elbasiony), sallam@f-eng.tanta.edu.eg (E.A. Sallam), tarek@ictp.org.eg (T.E. Eltobely), mfn_288@hotmail.com (M.M. Fahmy).
[1] Tel.: +20 1272503682.
[2] Tel.: +20 1000078006.
[3] Tel.: +20 1006209937.

Peer review under responsibility of Ain Shams University.

2090-4479 © 2013 Ain Shams University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asej.2013.01.003

represent a specific attack [2,3]. Anomaly detection discovers attacks by identifying deviations from normal network activities [4,5]. Misuse detection can discover attacks with a low false positive rate, but it cannot discover novel attacks. Anomaly detection can discover novel attacks but with a high false positive rate [2]. To avoid disadvantages of misuse and anomaly detection techniques and maximize their advantages, there are a lot of proposed hybrid approaches [6,7].

Many current IDSs are rule-based systems that depend on a set of rules representing attacks or normal network characteristics, which are collected and identified by security experts [8]. There is no doubt that manually rule encoding is a very expensive process in both time and money, moreover it depends on the efficiency of human experts in analyzing a huge amount of network activities to discover intrusion patterns. However, these drawbacks are overcome by employing many data-mining techniques in IDSs [6,9,7,10–13]. Data mining is the analysis of (often large) observational datasets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [14]. Thus, data mining techniques can be used to classify network connections into intrusion and normal data based on labeled training data in misuse detection [7], and to group similar network connections together in clusters according to a given similarity measure in anomaly detection [15,16]. This paper proposes a hybrid IDS based on two famous data mining algorithms called random forests [17], and k-means. In the misuse detection part of our framework, random forests algorithm is used to classify network connections into intrusion and normal data based on a labeled training dataset that helps it in building classification patterns. In the anomaly detection part, k-means algorithm is used to group network connections data into K clusters based on the similarity of connections features, then, some clusters of them are chosen as anomaly clusters.

There are many challenges in both misuse and anomaly detection parts, one challenge is the imbalance between intrusion types in real network connections datasets which are used as training data to our misuse detection system. Some types of intrusions like denial of service (DoS) have many network connections than other intrusion types like user to root (U2R); so, any data mining approach will be interested in decreasing the overall error rate of the system regardless of intrusion types, which causes increasing the error rate of the minority attacks like U2R, although these attacks are very dangerous than majority attacks [7].

Another challenge of anomaly detection is that network connections dataset contains important categorical features, while the k-means algorithm can only work with continuous features; moreover, the other continuous features are on different scales, which deceive the algorithm causing bias toward the highest scale features over the other features. In this paper, solutions for these problems are proposed.

On the other side, connections features do not have the same effect on the type of connection in terms of being normal or intrusion; so, some methods are proposed for feature selection and weighting that increase the detection rate of the proposed anomaly detection method [7,11].

One of the problems of anomaly detection is determining the anomalous and normal clusters after clustering all data. There are two assumptions that the majority of real network activities are normal, and the intrusion activities are not consistent with the other activities [10], however, these assumptions are not always true because of the high degree of similarity between some kinds of intrusions and real activities, which makes these intrusions to stick to normal data in the same clusters causing very high false positive rates. In this paper, a supervised methodology is proposed to improve false positive rates of the proposed anomaly detection method.

The proposed methods are evaluated over a real network connections data which are generated from the Defense Advanced Research Projects Agency (DARPA) network connections, which was prepared by ACM Special Interest Group on Knowledge Discovery and Data Mining in the Knowledge Discovery and Data Mining 1999 (KDD'99) contest [18,19]. All the algorithms are implemented in C#.Net based on the original implementation of these algorithms. The proposed work can be summarized as follows:

1. Employ the random forests algorithm in misuse intrusion detection as proposed in [7].
2. Employ the k-means clustering algorithm in anomaly detection.
3. Combine misuse and anomaly detection into a hybrid framework, and improve the overall performance by using useful output data from the misuse detection part like variable importance and some known intrusions, as inputs to the weighted k-means (wk-means) algorithm in the anomaly detection part.

The paper is organized as follows. Section 2 presents the misuse detection and its evaluation experiments on KDD'99 dataset, Section 3 presents the anomaly detection and its evaluation experiments on KDD'99 dataset, Section 4 presents the proposed hybrid framework and its evaluation experiments on KDD'99 dataset, finally, Section 5 summarizes the paper and discusses future work.

2. Misuse detection

2.1. Method description

The misuse detection method (see Fig. 1) employs the random forests algorithm as a data mining classification algorithm [7]. Like all supervised learning techniques, the method operates in two phases: training phase and classification phase. The first phase works offline to build intrusion and normal patterns based on a training dataset, the second phase works online to detect network intrusions based on the patterns generated from the first phase.

In the offline phase, a labeled training dataset is provided after some preprocessing into random forests based intrusion pattern builder, which builds the intrusion patterns needed for detection. In the online phase, network packets are captured by network sensors and converted to a network features database after some preprocessing. Then, the misuse detector classifies the network features database to normal and intrusions based on the patterns generated previously in the offline phase. If an intrusion is found, the alarm system raises an alarm.

Different intrusions do not produce the same number of network connections; the majority intrusions like DoS produce much more connections than minority intrusions like U2R. The imbalance nature of real network intrusions increases the classification sensitivity to the majority intrusions and decreases

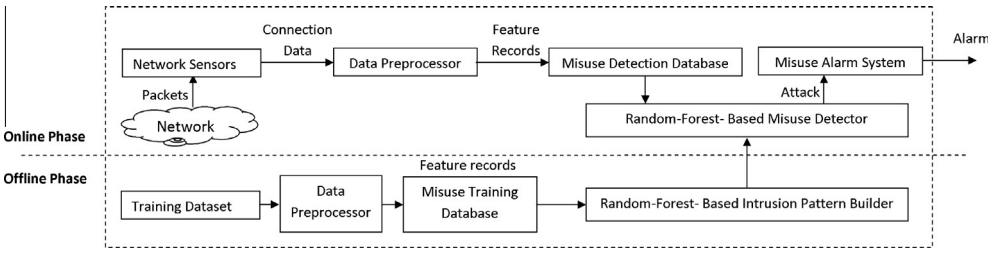


Figure 1 Misuse detection approach.

it to the minority intrusions although they are more dangerous than the majority intrusions [7]. As a solution to this problem, a balanced training dataset is made by downsampling the majority intrusions and oversampling the minority intrusions to increase their weights and decrease their error rate.

2.2. The random forests algorithm

The random forests algorithm is a classification algorithm consisting of a collection of tree structured classifiers, where each tree casts a unit vote for the most popular class at each input [17].

Each tree is grown as follows:

1. If the number of cases in the training set is N, a sample of N cases is taken at random from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number m M is specified such that at each node, m variables are selected at random out of the M input variables, then, the best split on these m is used to split the node. The value of m is held constant during the forest growing.

Each tree is grown to the largest extent possible. There is no pruning [17].

2.3. Experiments and results

The KDD'99 datasets are used as training and test sets to achieve our experiments. These datasets are 41 extracted features data from DARPA tcpdump data in 1998 [19]. KDD'99 datasets consist of three datasets; the first one is the full training set which has 4,898,431 connection records. The second one is the 10% training set which was taken from the full training set, it has 494,021 connection records. The third one is the test set which has 311,029 connections data. These datasets include normal connections in addition to four main types of intrusions, probe, DoS, U2R, and R2L with different connection counts.

As mentioned above, the 10% training set is used to construct a balanced training set by downsampling DoS and Normal connections, and oversampling U2R and R2L connections; so, random 10% connections of Normal and DoS types from the original 10% dataset are used, and also U2R and R2L connections are oversampled by replicating their connections. The balanced training set consists of 60,620 connections [7]. Table 1 lists the counts of connection types in all datasets.

There are two important parameters that significantly affect the error rate of the random forests algorithm: the number of

Table 1 Counts of network connections in KDD'99 and balanced datasets according to connections main types.

Dataset	Normal	DoS	Probe	U2R	R2L
Full dataset	972,781	3,883,370	41,102	1126	52
10% Dataset	97,278	391,458	4107	1126	52
Balanced dataset	9728	39,146	4107	4504	3135
Test dataset	60,593	229,853	4166	16,189	228

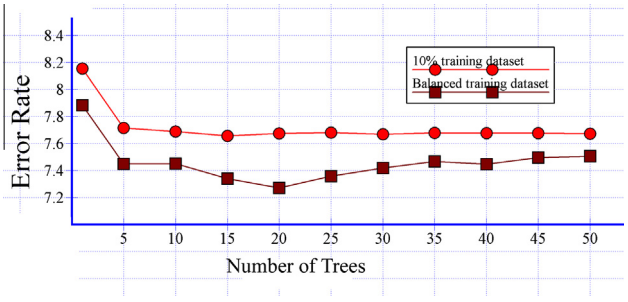


Figure 2 Test error rates using 10% and balanced training datasets.

random features used to split each tree node ($Mtry$), and the number of trees grown in the forest (Jbt). To achieve a good performance and a low error rate, these parameters are optimized by building the forest with different $Mtry$ and Jbt values, testing these forests using KDD'99 test set, and choosing the best values that achieve the lowest error rate, which were $Mtry = 20$ and $Jbt = 20$. Ver. 5.1 implementation of random forests algorithm in [17] is used to do our experiments.

The efficiency of the balanced training set is tested versus the 10% training set by applying them to our method as training sets and testing the built patterns on the KDD'99 test set, the results in Fig. 2 shows that the balanced dataset is more efficient than the 10% dataset.

Fig. 2 shows that the best error rate is 7.27% which achieved at 20 trees in the forest, with false positive rate of 0.54%, and detection rate of 91.23%. This error rate is better than the best KDD'99 contest which was 7.29% [18].

3. Anomaly detection

3.1. Method description

The proposed anomaly detection method (see Fig. 3) employs the k-means algorithm as a data mining clustering algorithm [20] to detect novel intrusions. It captures the network connec-

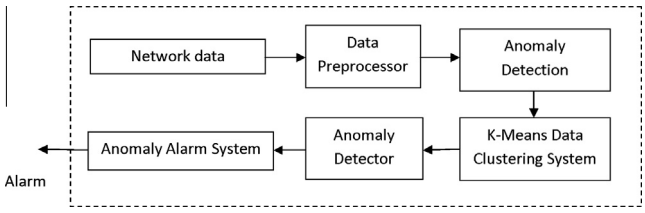


Figure 3 Anomaly detection approach.

tions data and converts it to an anomaly detection dataset by preprocessing, then, data are partitioned into homogeneous clusters using the k-means algorithm. After that, the anomaly detector determines the anomalous and normal clusters. The system raises anomaly alarm when anomaly clusters are detected.

3.2. The k-means algorithm

The k-means algorithm is a simple iterative method to partition a dataset into a specified number of clusters, K [20]. The algorithm is initialized by picking random K points as the initial K clusters "centroids", then, the algorithm iterates between two steps till convergence:

Step 1: assignment of each point to its closest centroid.
Step 2: relocation of each centroid to the mean of its assigned points. The k-means algorithm uses the Euclidean metric to quantify distance between points [20].

Suppose we have a dataset I having N entities, set of M features, V, and an entity-to-feature matrix $Y = (y_{iv})$ where y_{iv} is the value of feature $v \in V$ at entity $i \in I$. The algorithm will partition the dataset I to K clusters where $S = \{S_1, S_2, \ldots, S_k\}$, S_k subsets are the output clusters. Each cluster S_k is represented by a centroid $C_k = (C_{kv})$, then, the criterion minimized by the algorithm is the sum of distances to every cluster centroid:

$$W(S,C) = \sum_{k=1}^{k}\sum_{i\in I}\sum_{v=1}^{M} s_{ik}(y_{iv} - c_{kv})^2 \quad (1)$$

where $s_{ik} = 1$ if $i \in s_k$, and $s_{ik} = 0$, otherwise [11].

3.3. Experiments and results

Our anomaly detection experiments are achieved based on the KDD'99 datasets. Four datasets are constructed; each one consists of 30,000 connections chosen randomly from the KDD'99 10% dataset. The percentage of intrusions injected in each dataset is different from each other, four new datasets are generated: 1%, 2%, 5%, and 10% datasets, where the percentages of intrusions to normal connections into the specified datasets are 1%, 2%, 5%, and 10% respectively.

(1) ***The problem of categorical features***: the KDD'99 datasets contains three categorical features; *protocol_type*, which defines the protocol of the connection, e.g. tcp, udp, etc., *service*, which defines the service of the connection, e.g. http, telnet, etc., and *flag*, which defines the normal or error status of the connection [19]. The problem is that the k-means algorithm supports only continuous features because it depends on distances between the input points calculated by the Euclidean distance metric; so, the three categorical features are encoded to binary-valued features depending on the values of these categorical features which are selected in the specified dataset. Table 2 shows the values of the 10% dataset's three categorical features which have to be encoded as additional binary features. Table 3 shows the number of features of all datasets before and after adding the binary encoded features.

(2) ***The problem of different scales of features***: the continuous features of the KDD'99 datasets are on different scales. This causes the k-means algorithm to bias to the highest scale features as it depends on the squared distance measures of the Euclidean metric; so, all the features of each selected dataset are normalized; thus, the maximum value of each feature is one.

The "KMlocal" implementation of the k-means clustering algorithm is used as the clustering system of our anomaly

Table 2 The proposed KDD'99 10% dataset's categorical features and their values that would be encoded to binary features.

Feature name	Feature values			
protocol_type	icmp	tcp	udp	
flag	OTH	REJ	RSTO	RSTOS0
	RSTR	S0	S1	S2
	S3	SF	SH	
service_type	aol	auth	bgp	courier
	csnet_ns	ctf	daytime	discard
	domain	domain_u	echo	eco_i
	ecr_i	efs	exec	finger
	ftp	ftp_data	gopher	harvest
	hostnames	http	http_2784	http_443
	http_8001	imap4	IRC	iso_tsap
	klogin	kshell	ldap	link
	login	mtp	name	netbios_dgm
	netbios_ns	netbios_ssn	netstat	nnsp
	nntp	ntp_u	other	pm_dump
	pop_2	pop_3	printer	private
	red_i	remote_job	rje	shell
	smtp	sql_net	ssh	sunrpc
	supdup	systat	telnet	tftp_u
	tim_i	time	urh_i	urp_i
	uucp	uucp_path	vmnet	whois
	X11	Z39_50		

Table 3 Number of dataset's features before and after adding the binary encoded features.

Dataset (%)	Number of features before encoding	Number of features after encoding
1	41	72
2	41	78
5	41	79
10	41	95

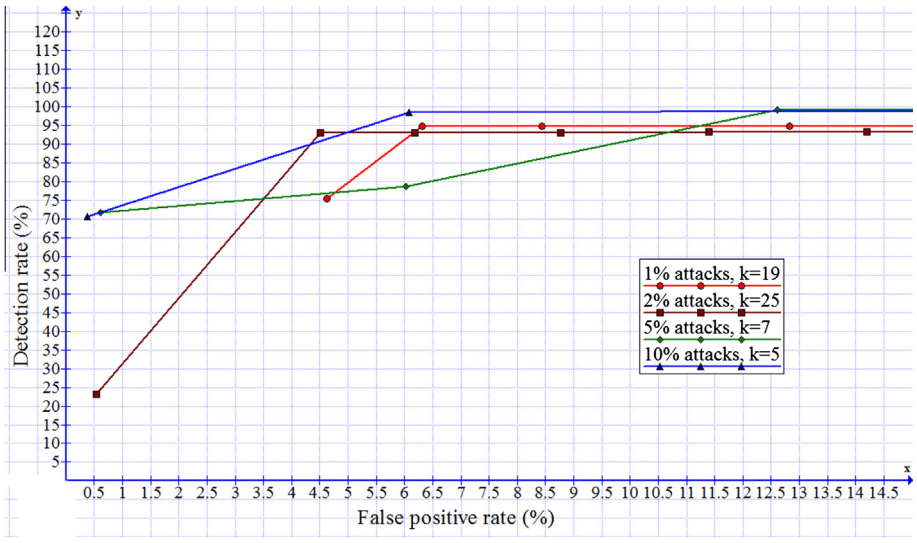


Figure 4 ROC curves for anomaly detection experiments using all generated datasets.

Table 4 A comparison between the proposed method and other methods in [4,7].

Approach	Remarks	Detection rate (%)	False positive rate (%)
The proposed method	1% Dataset	95	6.3
	2% Dataset	93	4.5
	5% Dataset	99	12.6
	10% Dataset	98.5	6
Method introduced in [7]	1% Dataset	95	1.5
	2% Dataset	95	2.5
	5% Dataset	90	2.5
	10% Dataset	88	2.5
Methods introduced in [4]	Cluster	93	10
	K-NN	91	8
	SVM	98	10

detection method [9]. The 1%, 2%, 5%, and 10% datasets are tested on the proposed method and the number of clusters (K) is optimized for each dataset.

Fig. 4 plots receiver operating characteristic (ROC) curves to show the relationship between the detection rates and the false positive rates over all generated datasets. The result shows that the method has a very high detection rate (reaches 98% in 5%, and 10% datasets), but still has the disadvantage of high false positive rate.

Ref. [4] proposed an unsupervised anomaly detection framework depending on outlier detection technique using three algorithms; cluster-based estimation, k-nearest neighbor, and one class support vector machine (one class SVM). Ref. [7] also proposed an unsupervised anomaly detection framework depending on outlier detection technique using the random forests algorithm, the frameworks proposed in [4,7] have been experimented on the KDD'99 datasets.

Table 4 makes a comparison between results of the proposed anomaly detection method and the methods in [4,7]. The results show that our method achieves more detection rate than the other methods especially in 5% and 10% datasets, but ours still has more false positive rate than methods in [4,7].

4. Proposed hybrid intrusion detection framework

According to our experiments of misuse and anomaly detection methods, the results show that anomaly detection method achieves high detection rate (reaches 99%) with a very bad false positive rate (reaches 12.6%), in contrast to the misuse detection method, which achieves somewhat bad detection rate (92.73%) with a very good false positive rate (0.54%); so, the proposed hybrid framework can strengthen the advantages of both misuse and anomaly detection by increasing the detection rate and decreasing the false positive rate.

The proposed hybrid framework consists of two phases: the online phase, and the offline phase. The online phase is a part of the misuse detection method; it is mainly responsible for comparing the network connections data to the generated intrusion patterns, if any intrusion is detected, a misuse alarm will be generated, and the attack features will be sent to the random attack selector component of the anomaly detection part. If the connection features do not match any attack, this connection is considered as uncertain data because it could be a novel attack, and will be sent to data preprocessor and merger component of the anomaly detection part in preparation for checking if it is an intrusion or normal connection. The offline phase contains the intrusion pattern generator of the misuse detection part, which uses the training dataset to build intrusion patterns used by the online phase. The pattern builder also outputs the feature importance values calculated by random forests algorithm used in the anomaly detection part. The offline part also contains all components of the anomaly detection part, which starts with merging the selected random attacks with the uncertain data from the misuse detection part, and creates the anomaly detection database using this data. After that, the clustering system depends on the wk-means algorithm as a data mining clustering algorithm used to cluster the connections data stored into the anomaly detection database.

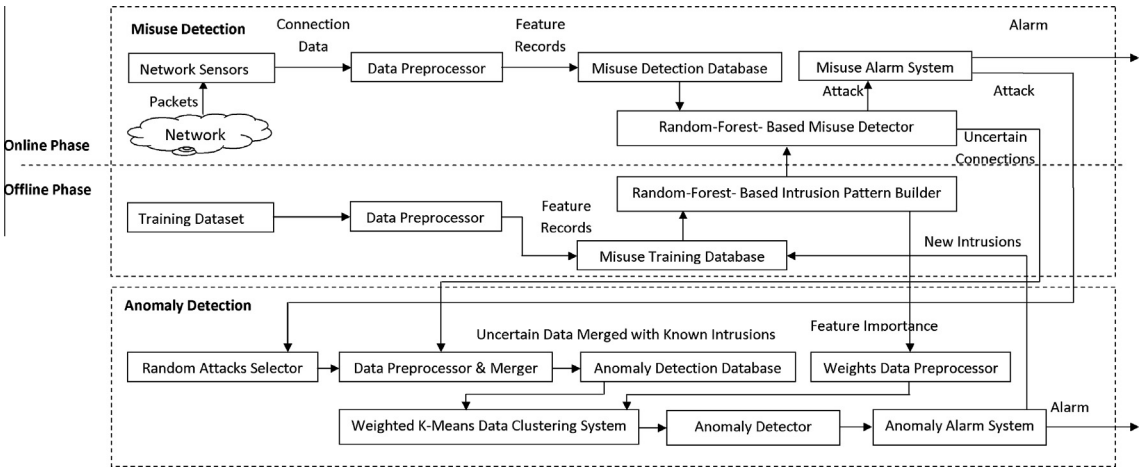


Figure 5 The proposed hybrid intrusion detection framework.

The anomaly detector component determines the anomalous and normal clusters by ordering the clusters according to the count of known intrusions into them, and select the clusters containing more known intrusions as anomalous clusters, finally, the anomaly alarm system component generates an alarm if anomaly clusters are detected, and send these anomalous connections to the training database of the misuse detection part to add these patterns to the intrusion patterns of it. Fig. 5 shows the proposed hybrid detection framework.

4.1. The weighted k-means algorithm

The wk-means is a modified version of k-means algorithm by including weights of the data features, so, criterion (1) is modified as:

$$W(S,C)=\sum_{k=1}^{K}\sum_{i\in I}\sum_{v=1}^{M}s_{ik}w_v^{\beta}(y_{iv}-c_{kv})^2 \tag{2}$$

where w_v is the weight of variable v, $v = 1, \ldots, M$, and w_v values are non-negative and sum to unity, and β is a user defined parameter [11].

k-Means algorithm assumes that all features have the same effect rate on the output class, but, according to our experiments in misuse detection, the random forests algorithm generates values of feature importance, which reflect the effect rate of all features on the correctness of output class labels, so, these feature importance values are taken and modified to satisfy the sum to unity condition by normalizing them and substituting the weights vector (w_v) by the normalized values in (2) at $\beta = 1$.

4.2. Feature importance measurements

As mentioned in the previous section, the variable importance measurements of the random forests algorithm are used to improve the clustering process efficiency. The random forests algorithm calculates the variable importance measures of all of the input features of the training dataset as described in [17]; to estimate the importance of the variable m, the number of votes for the correct class is counted using the out-of-bag (OOB) cases in every tree. Then, the number of the correct votes is counted again after randomly permuting the values of variable m in the OOB cases. The average of the margin be-

The procedure of generating and using the feature importance values

(1) For each feature m in the training dataset {
(2) For each tree t in the forest {
(3) Apply the OOB data to the current tree and count the votes of the correct class
(4) Randomly permute the values of the variable m in the OOB data.
(5) Apply the permuted OOB data to the current tree and count the votes of the correct class
(6) Subtract the number of the correct class votes of the permuted OOB data from the number of the correct class votes of the non-permuted OOB data
(7) The raw score of the importance of feature m is the average of these numbers over all the trees}
(8) Do the same for all features in the training dataset to obtain the importance vector of all features w
(9) Normalize the values of the w vector by dividing each value on the sum of all values of it
(10) Use the normalized vector w to substitute the values of the weights vector w_v in the wk-means algorithm

Figure 6 The procedure of generating and using the variable importance measurements.

Table 5 Output of clustering process of the 1% dataset, ordered by total connections count, using the k-means based anomaly intrusion detection.

Cluster no.	Total connections	Normal connections	Intrusion connections
1	89	89	0
2	130	130	0
3	237	237	0
4	279	279	0
5	370	370	0
6	402	176	226
7	481	481	0
8	565	507	58
9	837	837	0
10	1303	1303	0
11	1347	1347	0
12	1353	1353	0
13	1448	1435	13
14	1618	1618	0
15	1854	1854	0
16	2299	2297	2
17	2602	2601	1
18	5735	5735	0
19	7051	7051	0

Table 6 Output of the clustering process of the 2% dataset, ordered by total connections count, using the k-means-based anomaly intrusion detection.

Cluster no.	Total connections	Normal connections	Intrusion connections
1	57	56	1
2	101	101	0
3	138	0	138
4	233	233	0
5	256	256	0
6	313	313	0
7	365	364	1
8	420	2	418
9	495	495	0
10	757	757	0
11	775	774	1
12	822	822	0
13	1188	1188	0
14	1260	1260	0
15	1279	1279	0
16	1458	1455	3
17	1596	1596	0
18	1621	1621	0
19	1630	1627	3
20	1677	1648	29
21	1753	1753	0
22	2137	2137	0
23	2776	2776	0
24	2954	2948	6
25	3939	3939	0

tween these two numbers over all the trees in the forest is the raw importance score for variable m. The raw scores of all input variables are collected and normalized by dividing them on the sum of all the scores to satisfy the sum of unity condition. Then, weights vector of the wk-means algorithm are substituted by the final computed variable importance scores. Fig. 6 shows the procedure of generating and using the variable importance measurements.

4.3. Anomalous clusters determination using the known attack injection method

The main job of the wk-means algorithm is to partition the network connection data into homogenous clusters based on similarity measures, however, it cannot determine which clusters are normal connections and which are anomalous one, so, many anomaly detection methods use the two assumptions in [10], that the majority of real network activities are normal, and the intrusion activities are not consistent with the other activities. These assumptions have many problems as mentioned before in the introduction of this paper. Tables 5 and 6 show the network connections distribution after clustering the 1% and 2% datasets using the k-means algorithm; as noticed from these tables, although one of the two assumptions stipulates that the majority of connections are normal and suggests that the smallest clusters should contain many intrusions, but, the smallest clusters contain nothing but normal connections because these connections are different subtypes of normal connections, thus, selecting them as intrusions causes a very high false alarm rates.

As a solution to this problem, a supervised method is proposed to help in the detection of anomalous clusters. This method depends on the fact that most intrusions are similar; so, the clustering algorithm should collect most of them in the same clusters. If some known intrusions detected from the misuse detection phase are injected into the uncertain data before clustering, these known intrusions will stick to the unknown ones in the same clusters giving us bright spots into the anomalous clusters which make it easy to detect the anomalous clusters with a very low false positive rate.

4.4. Experiments and results

The KDD'99 datasets and the balanced dataset are used to do our experiments. The balanced dataset is used as a training dataset to the misuse detection part, and the KDD'99 test

Table 7 Values of the calculated weights using the random forests variable importance measurements of the balanced dataset.

Feature name	Calculated weight	Feature name	Calculated weight
duration	0.01289646	is_guest_login	0.005298988
protocol_type	0.102066446	count	0.104087754
service	0.079947128	srv_count	0.067050711
flag	0.065935818	serror_rate	0.002227385
src_bytes	0.023886258	srv_serror_rate	0.001909586
dst_bytes	0.126299029	rerror_rate	0.004283414
land	0	srv_rerror_rate	0.000373178
wrong_fragment	0.000647318	same_srv_rate	0.011693306
urgent	4.68348E−05	diff_srv_rate	0.018712547
hot	0.015209777	srv_diff_host_rate	0.003599064
num_failed_logins	0.000426924	dst_host_count	0.038294572
logged_in	0.102314793	dst_host_srv_count	0.067122974
num_compromised	0.002354455	dst_host_same_srv_rate	0.024671172
root_shell	0.004603246	dst_host_diff_srv_rate	0.027562406
su_attempted	0	dst_host_same_src_port_rate	0.052873232
num_root	0.000687205	dst_host_srv_diff_host_rate	0.008309154
num_file_creations	0.00243619	dst_host_serror_rate	0.006863946
num_shells	0.00015512	dst_host_srv_serror_rate	0.005994019
num_access_files	5.43733E−05	dst_host_rerror_rate	0.007453526
num_outbound_cmds	0	dst_host_srv_rerror_rate	0.001639974
is_host_login	0		

Table 8 Connections distribution of the first eight clusters of the 1% dataset, ordered by the counts of injected known attacks, using the wk-means based hybrid intrusion detection.

Cluster no.	Total connections	Normal connections	Unknown intrusions	Known intrusions
1	442	0	224	218
2	614	470	71	73
3	391	386	1	4
4	4119	4118	0	1
5	3506	3505	0	1
6	278	277	0	1
7	871	870	0	1
8	2300	2297	2	1

Table 9 Connections distribution of the first eight clusters of the 2% dataset, ordered by the counts of injected known attacks, using the wk-means based hybrid intrusion detection.

Cluster no.	Total connections	Normal connections	Unknown intrusions	Known intrusions
1	630	0	418	212
2	556	302	169	85
3	1650	1645	3	2
4	713	711	1	1
5	1663	1662	0	0
6	2238	2237	1	0
7	208	208	0	0
8	418	418	0	0

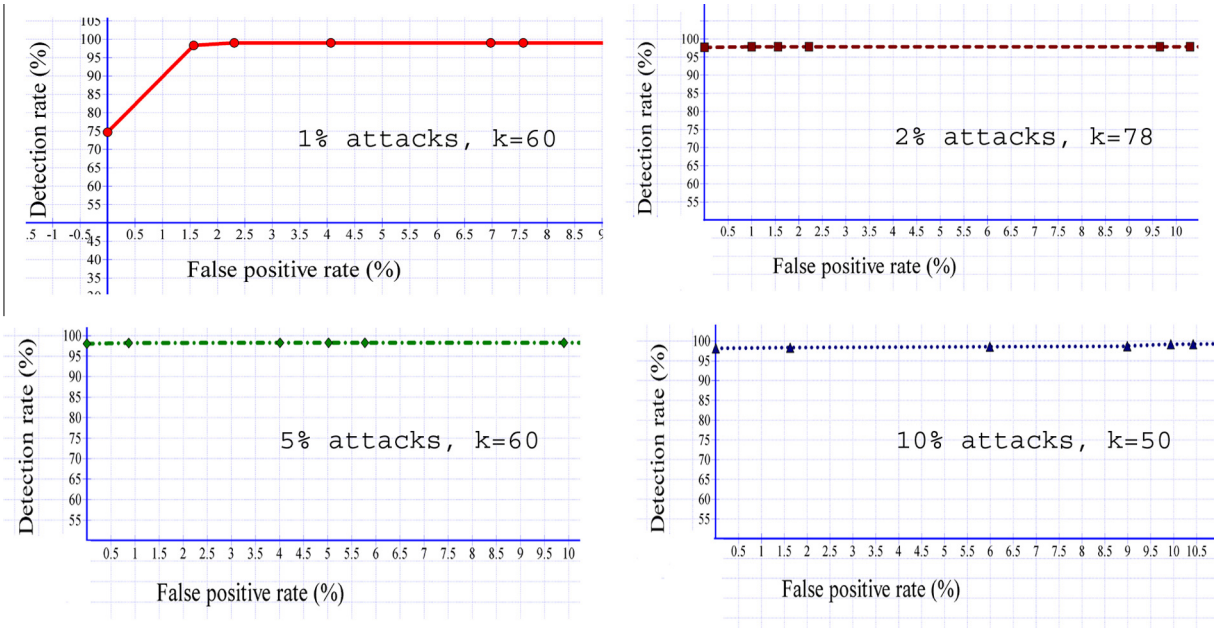


Figure 7 ROC curves for the proposed hybrid framework experiments using all generated datasets.

dataset as the captured network connections data. The "KMlocal" implementation of the k-means algorithm [9] is used after some modifications to support wk-means algorithm. The feature importance measures of the random forests algorithm in the misuse detection phase are modified as discussed before to be used as the weights vector for wk-means algorithm. Table 7 shows the values of the weights of the balanced dataset. The results of the misuse detection phase are discussed before, however, to cover most of the possible expectations, the output of this phase is changed manually to test the efficiency of the overall framework under many circumstances, so, the output of the first phase is supposed to be one of the following datasets, 1%, 2%, 5%, and 10% datasets, which means that the misuse part was not able to detect 1%, 2%, 5%, and 10% of intrusions, then, these datasets are created by randomly labeling some intrusions as detected or not detected as needed.

The problems of categorical and different scales features are solved as discussed in Section 3. The random attacks selector is adjusted to select 1% of the total connections, from the de-

Table 10 A comparison between results of the proposed misuse, anomaly, and hybrid detection approaches.

Approach	Remarks	Detection rate	False positive rate
The proposed misuse detection approach	KDD test set	92.73	0.54
The proposed anomaly detection approach	1% Dataset	95	6.3
	2% Dataset	93	4.5
	5% Dataset	99	12.6
	10% Dataset	98.5	6
The proposed hybrid framework	1% Dataset	98.3	1.5
	2% Dataset	97.8	1
	5% Dataset	98.2	0.9
	10% Dataset	98.3	1.6

Table 11 A comparison between results of the proposed framework and the framework introduced in [7].

Approach	Remarks	Detection rate (%)	False positive rate (%)
The proposed hybrid framework	1% Dataset	98.3	1.5
	2% Dataset	97.8	1
	5% Dataset	98.2	0.9
	10% Dataset	98.3	1.6
Framework introduced in [7]	KDD'99 test set	94.7	2

tected intrusions by the misuse detection part, as known intrusions, and then merges it with the prepared uncertain data.

Tables 8 and 9 show parts from network connections distribution after clustering the overall datasets; as noticed from these tables, the first clusters contain most of the intrusions contained in the datasets, thus, selecting them as intrusions will not cause high false alarm rates like the discussed anomaly detection approach.

Fig. 7 shows the results of our experiment on the proposed hybrid framework. The results show that the proposed hybrid framework achieves very high detection rates, reaching 98%, at very low false positive rates, 1.5% nearly.

Table 10 shows a comparison between the results of the proposed misuse, anomaly, and hybrid detection approaches. The results show that the proposed hybrid framework achieves high detection rates with low false positive rates compared to misuse and anomaly detection.

Table 11 shows a comparison between the results of the proposed framework and the results of the random forests-based hybrid framework in [7], the framework has been experimented on the KDD'99 datasets.

5. Conclusion and future work

This paper discusses the data-mining-based network intrusion detection systems. Two data-mining techniques are used in misuse, anomaly, and hybrid detection. First, the random forests algorithm is used as a data mining classification algorithm into a misuse detection method to build intrusion patterns from a balanced training dataset, and to classify the captured network connections to the main types of intrusions due to the built patterns. Our method is implemented in C#.NET by using the random forest original implementation [17] and tested through the KDD'99 datasets [7]. The main drawback of the misuse detection method is that it cannot detect novel intrusions that are not trained on before. Secondly, the k-means algorithm is used as a data-mining clustering algorithm into a proposed unsupervised anomaly detection method to partition the captured network connections into a specified number of clusters, and then detect the anomalous clusters depending on their features [10]. The "KMlocal" implementation of the k-means clustering algorithm [9] is used to implement our anomaly detection method. Our method is evaluated over the KDD'99 datasets after solving the problems of categorical and different scales features. The main drawback of the anomaly detection method is the high false positive rate. Thirdly, the random forests algorithm is used with the wk-means algorithm to build a hybrid framework to overcome the drawbacks of both misuse and anomaly detection. Feature importance values calculated by the random forests algorithm are used in the misuse detection part to improve the detection rate of the anomaly detection part. A supervised method is proposed to improve the anomalous cluster determination by injecting known attacks into the uncertain data before being clustered, and using these known intrusions in determining the anomalous clusters. Our experiment is evaluated on the KDD'99 datasets. The results show that the hybrid framework achieves detection rates and false positive rates better than the techniques in [4,7].

In the future, more advanced data-mining technologies like stream data mining could be used to increase the operations speed and make all processes online. In addition to that, the proposed attack injection method needs more study in the effect of changing the ratio of attacks to the uncertain connections dataset. It needs also more experiments on different real networks are needed to make sure of the efficiency of the proposed framework. Methods for determining the optimal number of clusters (K) in the k-means algorithm are also needed because its performance depends strongly on it.

References

[1] CSI/FBI Computer Crime and Security Survey. Computer Security Inst., San Francisco, CA; 2004. <http://www.issa-sac.org/docs/FBI2004.pdf>.

[2] Cole E, Krutz R, Conley JW. Network security bible. Wiley Publishing, Inc.; 2005.

[3] Bivens A, Embrechts M, Palagiri C, Smith R, Szymanski B. Network-based intrusion detection using neural networks. In: Proc Artif Neural Netw Eng, vol. 12, St. Louis, MO; 2002. p. 527–35.

[4] Eskin E, Arnold A, Prerau M, Portnoy L, Stolfo S. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Applications of data mining in computer security. Norwell (MA): Kluwer; 2002.

[5] Ramadas M, Ostermann S, Tjaden B. Detecting anomalous network traffic with self-organizing maps. In: Proc recent adv intrusion detect (RAID), lecture notes in computer science, vol. 2820, Pittsburgh, PA; 2003. p. 36–54.

[6] Barbara D, Couto J, Jajodia S, Popyack L, Wu N. ADAM: detecting intrusions by data mining. In: Proc 2nd annu IEEE workshop inf assur secur, New York; 2001. p. 11–6.

[7] Zhang J, Zulkernine M, Haque A. Random forest-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics – Part C. Applications and Reviews 2008;38(5):648–58.
[8] Snort Network Intrusion Detection System; 2006. <http://www.snort.org>.
[9] Mount D. KMlocal: a testbed for k-means clustering algorithms; 2005. <http://www.cs.umd.edu/~mount/Projects/-KMeans/kmlocal-doc.pdf>.
[10] Leung K, Leckie C. Unsupervised anomaly detection in network intrusion detection using clusters. In: Proc 28th Australasian CS conf, vol. 38, Newcastle, Australia; 2005. p. 333–42.
[11] Cordeiro de Amorim R, Mirkin B. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering,. Elsevier's J Pattern Recognit 2012;45:1061–75.
[12] Abraham T. IDDM: intrusion detection using data mining techniques. In: DSTO electron Surveill Res Lab, Tech Rep DSTO-GD-0286, Salisbury, Australia; 2001.
[13] Verwoerd T, Hunt R. Intrusion detection techniques and approaches. Comput Commun 2002;25:1356–65.
[14] Hand D, Mannila H, Smyth P. Principles of data mining. Cambridge (MA): MIT Press; 2001.
[15] Huang JZ, Xu J, Ng M, Ye Y. Weighting method for feature selection in k-means. In: Liu H, Motoda H, editors. Computational methods of feature selection. Chapman & Hall/CRC; 2008. p. 193–209.
[16] Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Elsevier's J Comput Security 2009;28:18–28.
[17] Breiman L, Cutler A. Random forests; 2006. <http://www.stat.berkeley.edu/~breiman/RandomForests/>.
[18] Elkan C. Results of the KDD'99 classifier learning. SIGKDD Explor 2000;1(2):63–4.
[19] DARPA Intrusion Detection Evaluation; 2006. <http://-www.ll.mit.edu/mission/communications/ist/CST/index.html>.
[20] Wu X, et al. Top 10 algorithms in data mining. Survey paper, Springer-Verlag London Limited 2007. Knowl Inf Syst, vol. 14; 2008. p. 1–37.

Reda Mohammed Elbasiony, Lecturer Assistant, Computer and Automatic Control Department, Faculty of Engineering, Tanta University, Egypt. The author received his M.Sc. degree in 2013 from the faculty of engineering, Tanta University. E-mail: reda@f-eng.tanta.edu.eg

Elsayed Adelhameed Sallam, Associate Professor, Computer and Automatic Control Department, Faculty of Engineering, Tanta University, Egypt. E-mail: sallam@f-eng.tanta.edu.eg

Tarek Elahmady Eltobely, Associate Professor, Computer and Automatic Control Department, Faculty of Engineering, Tanta University, Egypt. E-mail: tarek@ictp.org.eg

Mahmoud Mohammed Fahmy, Professor Emeritus, Computer and Automatic Control Department, Faculty of Engineering, Tanta University, Egypt. E-mail: mfn_288@hotmail.com